EXHIBIT A

Samir Kelekar

From: "ramdass keshavamurthy" <ramdassk@hotmail.com>
To: "Samir Kelekar" <graceful@vsnl.net>
Cc: "Muralidhara K." <murali@peertone.com>
Sent: Friday, January 03, 2003 3:47 PM
Attach: sec_assure_biz_plan2.zip
Subject: My comments on your revised business plan.

Hi Samir,

I have done the first level review of your business plan. I have set the tracking changes on and made the corrections, mostly typographical.

In addition to that, I have the following recommendations (changes):

- I have modified the title to indicate "assurance" from the word go. Whether automated makes sense once u put assurance, you can decide.
- I have removed any long term projections (like 4 years). It may not be required at this stage.
- I feel it is a good idea to introduce the product as a suite (va tool, autofixing and tracking agent).
- In addition, it is a good idea to indicate the client and server components. For eg., where each tool sits. For eg., the VA tool might be sitting at the server and the (tracking and autofixing tools) would be sitting at the client.
- A deployment diagram done using something like "Visio" would be helpful. I can help you on that if you have difficulty working with visio.
- I feel it is good to avoid even mentioning building of a VA tool. You should just state only the integration with the existing tools.
- It is a good idea to put a disclaimer "that those vulnerabilities which can be fixed easily" will be taken up for autofixing in the first phase of development. Make a provision to leave out the difficult ones.
- It is a good idea to take a man month as 20 days (5 X 4) and not 25 days which is higher than (6 X 4).

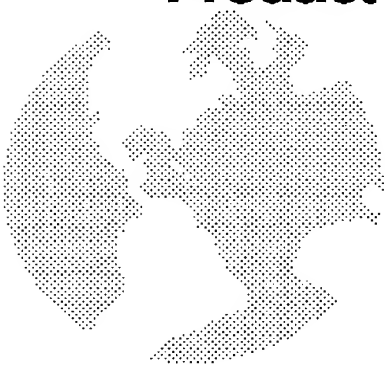
These are my comments. I have roughly glanced at your investment nos. I presume you have considered higher management costs and averaged it over the employees.

Ramdass

EXHIBIT A

EXHIBIT A

Towards Autonomous Systems for Network Security Assurance: A Product Business Plan



Submitted to :

Submitted by : *Dr. Samir Kelekar & team*

Author : *Dr. Samir Kelekar*

2 January 2003

EXHIBIT A

EXHIBIT A

Table Of Contents

1 **Executive Summary** 5

2 **Introduction** 6

3 **Market for network security** 7

4 **The problem and the solution** 8

4.1 The Problem of network security 8

4.2 Solutions: existing and proposed 9

5 **The Product** 10

6 **Competition** 11

7 **More details of the Product** 12

7.1 The VA component 12

7.2 The auto-fixing/patch management component 13

7.3 The Services-tracker component 14

7.4 Other Features 14

8 **Pricing** 16

8.1 Pricing of related security products 16

8.2 Our pricing 16

9 **The plan** 17

9.1 The scope of work 17

10 **Summary of Costs and Revenue** 22

11 **Funding Needed** 23

12 **Marketing** 24

13 **Annexure – Profile of the Team** 25

EXHIBIT A

1 Executive Summary

I. We are a group of network security professionals with immense experience in the IT industry. Two of us have Ph.Ds from top US universities with experience in some of the top firms, and a third has tremendous successful entrepreneurial experience.

II. This document is our business plan for a network security assurance[1] product --- one that will ensure that the network services running on your system are secure all the time.

III. This is a niche area in which there are no direct competitors to the best of our knowledge.

IV. The amount of immediate funding required is US $1.1 million. This money will be used for development of the product, as well as setting up the infrastructure, as also initial customer reach.

V. We expect to generate a revenue of $300,000/- in the first one and a half years.

VI. With very strong IP and initial customers, we expect the worth of the company to be at least $10 million at the end of the first one and a half years.

~~VII. We expect to break even in about 4 years, and we could provide details of the business plan for the rest of the 4 years.~~

[1] Security is assured to the extent of current state of the art, and certain other conditions. Adequate legal liabilities will be put in here so that the 'assurance' aspect does not get misused.

EXHIBIT A

2 Introduction

This document describes the business plan for a security assurance product for enterprise systems and networks. The kind of product described here would be an essential building block in future autonomous systems --- systems that correct themselves whenever a fault (in this case a security hole) is found in the systems. The market projections, the product, details of the competition, as well as the financials for the first one and a half year are described in the following sections.

EXHIBIT A

3 Market for network security

Even during the times of the Internet bust and an overall IT industry recession, the field of network security is booming. From a one-off firewall that companies used to deploy a few years back, many companies now have a full-fledged security administration division --- along with system, and network administration –and have a deployment of a veritable array of products: authentication, and encryption packages, intrusion detection systems, firewalls, system hardeners, wrappers, logging infrastructure. It is an accepted fact that without an adequate security infrastructure, issues of security breach outweigh the benefits of the Internet and business continuance is impossible. Security services and consultants are in hot demand under the current scenario and current rates per hour for a security consultant in the US are equal to or upwards of $90/- an hour[i]

According to Lehman Brothers estimates[ii] the Internet Security software industry is forecasted to be over $13.3 billion by 2004, equating to a 24% growth rate.

Specifically, the market for patch management, a component of the product described here, is currently $2 billion per year. In a recent survey of users whereby they gave their wish list for Microsoft, what topped was the need to have a good system for patch management and distribution for Microsoft products. It is estimated that U.S. companies currently spend around $2 billion a year on patch management alone, and 80% of all security breaches could be avoided if companies are prompt in deploying available patches on systems.

4 The problem and the solution

4.1 The Problem of network security

There are many aspects to security that are important. To begin with, we all know of anti-virus tools such as Norton which sit on your Windows based systems and ensure that these systems are free of viruses all the time. Anytime a new virus is discovered, Norton develops a virus detector and blocker and adds it to your system automatically (you give a go/no-go for the update) thus ensuring that your system is secure from the newly detected virus.

A more difficult problem that is faced by security administrators today is to ensure that your system is secure not just from viruses but from newly discovered security vulnerabilities in the network services that are being run on the system. For instance, nowadays, hackers find and exploit security holes in network services. Hackers are increasingly up-to-date with the state of the art, and many a times hack into a system within hours of a new security vulnerability being discovered. Thus, one of the problems a security manager or administrator is faced with is to patch or appropriately reconfigure the system to be secure when a security vulnerability is found, before malicious hackers are able to exploit the vulnerability.

Even doing this does not solve all the worries of a security administrator. It is always possible that some of the network services run are either configured to be non secure or their configuration is changed as part of the running of the system, thus making a previously secure system now vulnerable. Consider for instance the following scenarios that happen quite commonly during the run of a system.

- A user installs and runs a chat client which is known to have vulnerabilities in it.
- A user opens a Windows SMB share with an easily crackable password. (Windows 2K would require administrator ~~privileges~~privileges for the above, but it is possible that even the administrator uses a weak password.)
- An insider (an employee of the company) deliberately rolls back a patch or starts a vulnerable service --- perhaps a backdoor on an open port, so that he or his friend can hack the system from outside.
- A user inadvertently configures his/her browser to accept hostile applets.

All of the above scenarios make a previously secure system vulnerable in the course of day-to-day operation.. Thus, the biggest problem security administrators are faced with is that of keeping one's system secure all the time; this involves protecting it from newly found as well as other previously discovered vulnerabilities, as well as protecting it from attacks that may exploit security holes created as a result of non secure configurations of

EXHIBIT A

services, as also from a change in configuration of a service from a secure one to a non secure, as well as from starting vulnerable services.

4.2 Solutions: existing and proposed

Today, VA tools are run off-line to discover vulnerabilities in a system. These vulnerabilities are then fixed either by deploying patches or configuring/reconfiguring systems. However, nothing ensures that the system is secure in the time period between two runs of a VA tool.

In other words, there is no tool out there which will ensure that the system can be made secure in case the scenarios described in the previous section do take place. There is no tool out there which detects vulnerabilities in your system are as they happen. Doing this in an automated fashion is what the security assurance tool is meant to do. In other words, this business plan is about developing a tool which is a generalization of Norton, one that will ensure that your system is free from all network based security vulnerabilities. The security assurance tool will sit on your system, all the time looking for security vulnerabilities on your system, fixing them if found (of course with an appropriate go/no-go from the security admin.), as well as updating your systems with required patches to keep your system secure.

The next section describes the product.

EXHIBIT A

5 The Product

The product idea is a straightforward one. The product will consist of a VA component, another component that fixes security holes automatically, and a third component that tracks services and their configuration files via hooks, and signals to the first two components when a particular configuration changes or a particular service gets started. The first two components then take over, figure out what new vulnerabilities have been created, and fix them automatically. **The aspect of tracking services all the time, and dynamically intimating the VA tool and fixing the vulnerabilities automatically is entirely new and is the real value-add in all this. This idea should be patentable.**

To describe the product in detail, to begin with, the enterprise network will be scanned using a VA tool and the holes found there in will be fixed automatically using a auto-fixing component and ~~and~~ a patch management component. An auto security fixing component fixes certain kind of vulnerabilities that a VA tool detects. This would include reconfiguration of services, blocking of certain ports etc. A patch management component fixes those holes which require vendor-supplied patches to fix.

Having done the above, the system is now secure. (Of course, we assume that there is no intrusion already present in the system.). The component that tracks services now becomes active on the system, in the same way as a virus-blocker sits on a Windows based PC continuously looking for viruses in emails and attachments. The moment a service that accesses a network port, or a service that could have a weak password or some other such potentially vulnerable service is started, or configurations of already running services are changed to make them potentially vulnerable, the services-tracker would intimate the VA tool; the VA tool would then run that part of the VA that pertains to the kind of vulnerabilities that the intimation has suggested. If the VA tool finds that a new vulnerability has been created, it will fix the vulnerability automatically with the help of the auto-fixing and patch management component. , Of course, enough safeguards such as getting a go/no-go from the administrator would be present depending on the policy that is set by the administrator. Thus, the idea is that the network will be secure all the time. Of course, there are certain conditions; namely, that there should be no intrusion in the system in the first place. Also, it is possible that hackers stretch the limits of technology by discovering previously undiscovered vulnerabilities and exploit them.

The next section describes the competition.

EXHIBIT A

6 Competition

Briefly there are a number of companies in the VA tools space. They include companies such as Cisco, ISS, eEye Digital Security, Symantec, NetIQ, Bindview, Network Associates, Harris as well as the open source VA tool called Nessus.

A recent addition to the market is the concept of patch management. These patch management tools interface to VA tools and install patches to fix those vulnerabilities that require patches. It is not clear if they fix vulnerabilities that do not require patches, though all indications are that they don't. Among the patch mangement companies are Patchlink, ConfigureSoft, BigFix, Shavlik Technologies, and Gibraltar Software. Patchlink, ConfigureSoft, BigFix, and Shavlik Technologies currently have patch management products for only Windows based systems, while Gibraltar concentrates on Linux patch management.

What these companies ***clearly do not do is security assurance*** (we have studied the Patchlink offering, and read the literature about the other competitors): in other words, these tools do not ensure that your system remains secure between two runs of their VA tools. Patchlink does have a patch-compliance feature wherein if a patch once deployed is rolled back, an alert is generated.

Thus, the space we are getting into is new. We have a USP. There is competition in related areas, but not exactly in what we are doing.

It is clear that there is a market need for such a product. One cannot afford to have a system non secure at any time. Current offerings from competitors do not ~~feel~~ fill this gap.

7 More details of the Product

The software product intended is one that will ensure security assurance: that is it will detect vulnerabilities as they happen and fix security holes automatically ensuring security assurance all the time; thus, it is one step towards making a system an autonomous (self-correcting) system, a concept IBM is pioneering in a number of areas.

An autonomous system works by taking action to fix problems that occur in a system automatically. In the security context, this means that once an allied component such as a vulnerability assessment tool indicates that it has found a security hole, the autonomous system (our product described below) takes over, and fixes the security holes. The VA tool is intimated of a potential vulnerability by hooks which run on the system and indicate whether new services that may be potentially vulnerably are started or ~~exisiting~~existing services are made vulnerable via reconfigurations. The autonomous system would then again run the concerned vulnerability assessment tool and would ensure that the holes are indeed fixed, as also it will check and make sure that there are no other unnecessary repercussions on the system.

A side product of this ~~produt~~product suite, would be a component that would be general enough to interface to standard proprietary vulnerability assessment systems, as well as open-source VA systems such as Nessus etc

7.1 The VA component

As an example of what is involved in developing the VA component of the product, Nessus the open-source VA tool currently tests for over 1,100 vulnerabilities.

Some of them come under the following categories:

- Backdoors
- CGI abuses
- CISCO
- Denial of Service
- Finger abuses
- Firewalls
- FTP
- Gain a shell remotely
- Gain root remotely

EXHIBIT A

- General
- Misc.
- Netware
- NIS
- Port scanners
- Remote file access
- RPC
- Settings
- SMTP problems
- SNMP
- Untested
- Useless services
- Windows
- Windows : User management

The idea is to come up with automatic fixes for all of the above categories. Secondly, we intend coming up with custom interfaces to all the standard vulnerability assessment tools in use, and develop automated security hole fixing as a core area. Our tool would fix holes for all standard operating systems, namely the various consumer Windows versions, Windows NT/2000, as well as Linux, and Solaris to begin with. Later the product can be extended to HP-UX, SCO and other versions of Unix.

7.2 The auto-fixing/patch management component

The auto-fixing and the patch management components would fix the security holes that are detected by the VA tool. The idea is to come up with automatic fixes for all the categories of vulnerabilities described above in the VA tools section. As a bye-product of the effort, we could come up with custom interfaces to all the standard vulnerability assessment tools in use, so that the auto-fixing/patch management components could be used as stand-alone components that interface to various VA tools.

Our auto-fixing/patch management component would fix holes for all standard operating systems, namely the various consumer Windows versions, Windows NT/2000, as well as Linux, and Solaris to begin with. Later the component can be extended to HP-UX, SCO and other versions of Unix.

EXHIBIT A

It is to be noted that the auto-fixing component that we have envisaged is fairly complex to develop. Consider a typical example of misconfiguration of a system from a security point of view. Sendmail is a popular SMTP service that is used to send and receive mail on Unix systems, one that has been a frequent target of hacking attacks; Consider that sendmail is misconfigured to allow relaying. Relaying allows other users to use your sendmail server to send his/her emails thus stealing your bandwidth; further, the user can also masquerade him/herself and send spam.

To reconfigure sendmail so that it is secure, one will have to do the following using scripting:

1. Figure out whether sendmail is running.
2. Get its process id, and shutdown sendmail. (Of course, one would require appropriate permissions to do so.
3. Parse the sendmail configuration file to figure out how it is configured, and what needs to be change.
4. Change configuration file of sendmail appropriately, backup original configuration file.
5. Start sendmail again.
6. Make sure there are no other repercussions on the system.
7. Run the releant vulnerability test again to make sure that the vulnerability is indeed fixed.

Other fixes might require stopping a particular service. This could for instance require reconfiguring the xinetd.conf file in Unix. Another fix could require returning a bogus version of a software to the outside parties so as to send a potential attacker on the wrong track. Other fixes could require downloading an ~~appopriate~~appropriate patch via the patch management component part of the tool.

7.3 The Services-tracker component

The key component in the product is the services-tracker component. This component tracks, using hooks, whenever a particular services that accesses a UDP, TCP port or one accessing the IP protocol is started, or if configuration files of such services are changed. It also tracks whether passwords of services that are accessible from the network are changed. This technology is being currently developed via a prototype and should be patentable.

7.4 Other Features

Other standard features that are needed for such products include auto-tracking of inventory on systems, as well as auto-updating of the patches, auto-fixing and service-tracking scripts to fix new vulnerabilities, when discoveries of such vulnerabilities get announced in various forums. The latter facility is extremely important in a field such as network security where technology moves ahead on a day-by-day basis.

EXHIBIT A

There are various other features such as use of peer-to-peer technology whereby a peer client learns from another peer client. Details of these aren't being describe in this business plan.

EXHIBIT A

8 Pricing

8.1 Pricing of related security products

Patchlink's patch management software Update costs $995/- for a server license, and then $1200/- for Unix client licenses (minimum 10 licenses require to be bought, each costing $120/- each), and $150/- for Windows licenses (minimum $15/- per license and a minimum of 10 require to be bought). (Their Unix client is not yet ready.)

Thus, for a small enterprise, one having say less than 10 Unix and 10 Windows machines, it would cost $2,345/- for running Patchlink's Update. Other VA tools --- those which do not have auto fixing facility --- cost upwards of $995/- per license. Nessus, the open-source tool, is free however.

8.2 Our pricing

Since our product does much more than either patch management or VA tool, our price would have to be different.
Given the price of a VA tool, the price of a patch ~~mangement~~management component in the market ,we could price our system at US $3000/- for the equivalent ~~liecnees~~licenses as above and still be competitive. Out of this, we could keep $500/- for the deployment licences of certain third party components that we may use in our product.

In rupee terms, this amounts to Rs. 1,50,000/- per license given a conversion of Rs. 50/- for a US dollar.

EXHIBIT A

9 The plan

9.1 The scope of work

This section describes the scoping of the work to develop the product described in the previous sections.
There are currently around 1100 vulnerabilities that a tool such as Nessus can detect.. A rough guestimate is that around 800 or so vulnerabilities could be fixed in an automatic manner as also that we would consider the same number for installing hooks on services that can determine where a service has been started leading to the above kind of vulnerability. (This is just a ballpark number, which may change after a detailed study of the nessus source code.)

The estimate to write a script to detect a vulnerability, a script to install a service –hook as well as fix a hole automatically (an instance is described in the previous section) is at least 3 staff weeks (15 staff days).

Given the above, for automatic fixing of 800 vulnerabilities, it could take around 12,000 staff days, which comes to 2400 staff weeks.

Given the enormity of the task, it would be better off to phase the work in the following manner.

Version 1.0: auto-detect, and fix top 100 vulnerabilities automatically
Version 1.1: auto-detect and fix top 400 vulnerabilities automatically
Version 2: auto-detect and fix top 800 vulnerabilities automatically.

The top vulnerabilities can be determined by their frequency of occurrences. SANS institute comes up with a list of top 20 vulnerabilities, for instance.

The table below gives the details of the development time involved for the above, given that we have 10 developers for the first version, going onto 30 for version 1.1, and increasing to 40 for version 2.

Time period	No. of vulnerabilities to be auto-deteeeddetected and fixed	Effort in staff days (1 vulnerability = 15 staff days)	No. of developers	Time in days	Time in months (1 month = 25 days)
Month 1 to 4	100	1500	10	150	6
Month 5 to 8	300	4500	30	150	6
Month 9 to 12	400	6000	40	150	6

EXHIBIT A

Thus, version 1.0 could be released in 6 months, version 1.1 at the end of 12 months, and version 2.0 at the end of 18 months. Of course, some additional development work namely for writing reporting tools as well as writing automatic updating of software via the web-site etc is required, a first-cut effort for which is given below.

The following table gives the number of developers required for other development work, namely writing reporting tools, writing automatic updating of the software via the web site.

Time period	No. of developers for writing reporting tools, auto. Update software
1 to 12 months	5

Given the above, the detailed plan envisaged for the firs 1.5 years is as follows: come up with version 1.0 in the first 6 months; the version will then be extended into version 1.1 in the next six months; marketing in the US would then begin. The idea is to have full product development in India, and only minimal marketing support in the US.

- March 2003- August 2004 --- Product Development in India
- September 2003 –August 2004 – Marketing in the US, and other countries including India, product support, and development of future versions in India. One could have some support based in the US.

Time Period	**Description**	**Costs (rupees)**	**Costs ($)**	**Description**	**Revenue (rupees)**	**Revenue ($)**
March 2003 – August 2004						
	Legal Costs	Rs.20,00,000/- (Rs. 20 lakhs)	$40,000/-			
March 2003 – August 2003	Development of Version 1.0 in India					
	Development Costs (15 persons with salary +	Rs.45,00,000/- (Rs. 45 lakhs)	$90,000/-			

EXHIBIT A

	overhead costs) Computed at Rs. 50,000 per staff month					
	Equipment and Software Costs	Rs.25,00,000/- (Rs. 25 lakhs)	$50,000/-			
September 2003 – February 2004	Development of version 1.1 in India					
	Development Costs (35 persons with salary + overhead costs)	Rs.1,05,00,000/- (Rs. 105 lakhs)	$210,000/-			
	Equipment and Software Costs	Rs.12,50,000/- (Rs. 12 and a half lakhs)	$25,000/-			
March 2004- August 2004 (The dates here overlap with the product development dates, but that is so that marketing and support efforts are distinct from product dev. efforts)	Operations in US as well as India			Sale of 100 licenses at $3000/- each	Rs.150,00,000/- (Rs. 1 crore 50 lakhs)	$300,000/-
	Cost of third party	Rs. 25,00,000/- (Rs. 25 lakhs)	$50,000/-			

EXHIBIT A

	deployment licences for 100 licences sold at $500/- per copy					
	Operations costs in the US (5 persons operations + overhead costs; costs taken at $15,000/- per staff month)	Rs.1,50,00,000/- (Rs. 1 crores 50 lakhs)	$300,000/-			
	Operations costs in India (45 persons with salary + overhead costs; total taken at $1000/- per staff month)	Rs.1,35,00,000/- (Rs. 1.35 crores)	$270,000/-			
	Other Marketing Costs	Rs.25,00,000/- (Rs. 25 lakhs)	$50,000/-			
	Total Costs for the first 1.5 years	**Rs. 5,42,50,000/- (Rs. 5 crores 42 lakhs 50 thousand)**	**$1,085,000/-**	**Total revenue for the first 1.5 years**	**Rs.1,50,00,000/- (Rs. 1.5 crores)**	**$300,000/-**

EXHIBIT A

EXHIBIT A

10 Summary of Costs and Revenue

The following table gives the summary of costs and revenue for the first year.

Year	Costs in US dollars	Comments about Costs	Revenue in US dollars	Comments
March 2003 – Feb 2004	**$1,085,000/-**	Product development, and marketing	**$300,000/-**	Sale of 100 licenses

EXHIBIT A

11 Funding Needed

The funding required is as follows.

- (For the period March 2003 – August 2004) = US $ 1.1 million

EXHIBIT A

12 Marketing

The tool will be sold through channel partners as also direct marketing wherever possible. VA tool vendors can sell this tool minus its VA component as a value-add. Managed security providers are an important customer base for this product. They could also potentially act as Beta customers whereby they offer the services offered by this tool to their customers. Tie-ups with security and other companies would definitely help.

EXHIBIT A

13 Annexure – Profile of the Team

Dr. Samir Kelekar has around 19 years of professional experience in the area of computing. He has a BTech from the Indian Institute of Technology, Bombay, India, a M.S. in Computer Engineering from Clemson University, Clemson, SC, USA and a Ph.D. in Electrical Engineering from Columbia University, New York, USA. During his career, he has worked with IBM T.J.Watson Research Center, New York, the Naval Postgraduate School, Monterey, California, Motorola India Electronics Limited, Bangalore, BFL Mphasis, Bangalore, Alcatel Internetworking, Bangalore and Sanware Storage Solutions, Bangalore. He has headed medium sized teams, and was head of Alcatel Internetworking's Network Management Software Division in Bangalore. He has also worked as a senior manager in the Bangalore division of a Silicon Valley storage start-up, Sanrise. He has published in international publications such as the IEEE/ACM Transactions on Networking, and IFIP's Protocol Specification Testing and Verification Symposium. He has worked in both research as well as development, as well as has experiencing managing projects. Currently, he does training and consultancy to corporates in the area of network security. He has also written for Express Computer in the area of network security.

Mr. Muralidhara K, has a Bachelors degree in Electronics and Communications engineering. He has expertise in the development of knowledge based systems. He is one of the founders of NetExpert Inc, a private company that developed artificial intelligence software. His team successfully took the company into the public markets in the USA by merger with Ultrexx Corporation, which is currently trading under the symbol ULXX on the bulletin board. Mr. Muralidhara was the principal developer of CruXpert, a knowledge based development tool. In addition, he was the chief architect of the Intelligent Bridge Framework (a tool for development of interactive knowledge banner ads) developed at Ultrexx. Prior to that he had worked in Hewlett Packard (India) Software Operations, National Semiconductors (USA) and Artificial Intelligence Lab, IISc, Bangalore. In addition to the above, he was also the principal architect of the flagship product CruX Dongle while being the Managing Partner at CruX Technologies. This product was designed, manufactured by CruX Technologies and marketed to over 40 ISV's(Independent Software Vendors) including many Multi-National corporations like TISL, IESL (ITI Equatorial Satcom Limited) Siemens etc. Further, he was involved in the design and pilot implementation of an Intelligent Configurator built using a Constraint Logic Programming (CLP) language for Allen Bradley, Rockwell Automation, Cleveland.

Dr. Ramdass K has a doctorate in Engineering from the **University of Southern California, Los Angeles.** He has extensive experience in the areas of Numerical Computation, Engineering and Client Server Software Development, Distributed Computing Methodologies and Expert Systems. His current focus is on agent based Distributed Architectures. Prior to founding this company he worked with **Ultrexx** Corporation, where he was the architect of the visual expert system environment called "Visual CruXpert". In addition, he was an active designer of the Knowledge Banner Initiative built around the Intelligent Bridge Framework. Earlier he was with, **EMRC** (http://www.emrc.com), a Detroit based computer aided design and development

EXHIBIT A

company. He was the Project Manager in charge of the development and support of many of the modules of the suite of Products from EMRC (called NISA). As a project manager, he led many application development and consultancy initiatives built around the NISA product suite. Prior to this, he worked as a Research Engineer at the Computer Aided Design Lab at IIT, Kanpur."

[i] Communication with a US based business consultant
[ii] Lehman Brothers report on the US security market March 7 2001